

# A Wideband, High-Resolution Spectrum Analyzer

M. P. Quirk

Institute for Defense Analyses, Princeton, New Jersey

H. C. Wilck, M. F. Garyantes, and M. J. Grimm

Communications Systems Research Section

*This article describes a two-million-channel, 40-MHz-bandwidth, digital spectrum analyzer under development at the Jet Propulsion Laboratory. The analyzer system will serve as a prototype processor for the sky survey portion of NASA's Search for Extraterrestrial Intelligence program and for other applications in the Deep Space Network. The analyzer digitizes an analog input, performs a  $2^{21}$ -point Discrete Fourier Transform, accumulates the output power, normalizes the output to remove frequency-dependent gain, and automates simple signal detection algorithms. Due to its built-in frequency-domain processing functions and configuration flexibility, the analyzer is a very powerful tool for real-time signal analysis and detection.*

## I. Introduction

A  $2^{21}$  (2,097,152)-channel spectrum analyzer is being designed at JPL for real-time processing of signals with bandwidths of up to 40 MHz from the radio antennas of the DSN. The spectrum analyzer will perform all functions from sampling the input waveform to detecting signals in the computed spectra. Applications include radio-astronomy spectral analysis and continuum studies, high-resolution frequency analysis of wideband receiver systems in the DSN, radio-frequency interference (RFI) studies, and spacecraft telemetry acquisition and tracking. The primary radio-astronomy application will be to serve as a prototype system for the sky survey analyzer for NASA's Search for Extraterrestrial Intelligence (SETI) program [3], in which the entire celestial sphere will be scanned in the microwave region for narrowband signals. The final system to be used for the survey is planned to have a bandwidth of about 300 MHz and will consist of segments based on the prototype design [7].

The use of high-speed, commercially available CMOS VLSI components will allow millions of spectral points to be computed every second. The real-time digital processing will be performed by a rack of special-purpose hardware modules configured as a pipe in which data flows from the input samplers through the Discrete Fourier Transform (DFT) to the frequency-domain processing modules. The requirements of the SETI sky survey have guided the design of the frequency-domain modules. The digital boards have parameters that can be programmed by a user to match specific signal processing and detection needs. Interaction with users and with other devices is through the system controller, a workstation computer that controls and tests the hardware.

The core of the analyzer is a matrix-style DFT pipe that can compute contiguous real-time transforms of up to two million points on a 40-MHz-bandwidth signal. The analyzer performs a long  $N = LM$ -point DFT by considering the input

data as an  $L \times M$  matrix, transforming the columns with  $L$ -long Fast Fourier Transforms (FFTs), multiplying the resultant matrix entries by complex rotation factors, and then transforming the rows with  $M$ -long FFTs (see Section IV). The input to the FFTs is a real sequence derived by mixing the Intermediate Frequency (IF) input to baseband and sampling at a rate equal to twice the IF bandwidth. Real rather than complex sampling is used in order to eliminate the imaging caused by mismatches between the anti-aliasing filters on the real and imaginary legs of a complex mixer. A  $2N$ -long real sequence is transformed using an  $N$ -long complex DFT by exploiting the symmetries in the spectrum. The output spectra are unscrambled and then processed by special-purpose modules which can be programmed to perform signal detection algorithms. For example, the Baseline module can be used to remove frequency-dependent system gain, to whiten the spectrum, or to suppress interfering signals. The Output Processor can pick peaks and averages of groups of frequency powers for display or further processing.

If desired, any board can perform a null operation and pass its input directly to its output. This allows testing of the boards in isolation or in combination with other boards in the pipe; it also allows flexibility in the operation of the analyzer. For example, by bypassing stages, transforms less than  $2^{21}$  points can be performed, processing stages can be skipped, and the analyzer output can be taken from any section of the pipe.

A special design feature is that the analyzer can process two separate input data streams in the pipe at the same time. An "input switch" signal controls the selection between the two inputs,  $A$  and  $B$ , and labels each spectrum flowing down the pipe as type  $A$  or type  $B$ . To accommodate two data streams, there are two power-accumulation memories, one for  $A$  spectra and one for  $B$  spectra. This capability was built in so that the analyzer could cope with Dicke switching, in which the antenna beam alternates between a source of interest and a background source for calibration. The addition of another IF stage would allow the analyzer to process any two input data streams; in particular, it may be desirable to process both the left-circular and right-circular polarizations of a radio-astronomy signal.

The analyzer pipe has two output valves for the real-time data: one for the spectrum imaging data and one for the output of the matched-filter detector. The output processor works as a valve for the spectrum imaging data, extracting coarse-resolution or narrowband data for the graphics displays. The other valve sends the workstation the frequency, bandwidth, and power of signals which pass a threshold test from the matched filter. The output from both valves can be used for postprocessing.

This spectrum analyzer is designed to be a flexible tool for real-time analysis at high data rates. In response to a user's needs, the system can be reconfigured by software in milliseconds. Configuration parameters which can be changed include the sampling rate, the length of the DFT, the accumulation frame lengths, the signal detection parameters, and the output processor controls.

The next section is a more detailed discussion of the spectrum analyzer system and its operation. Section III explains the synchronization and configuration controls. Sections IV and V describe the design and operation of the  $2^{21}$ -point transform, and Section VI describes the real-time frequency-domain processing modules and how they will be used for the SETI sky survey.

## II. The Spectrum Analyzer System

The spectrum analyzer is a pipeline architecture in which, after a start-up delay, all stages of the algorithm are executing concurrently [1]. Figure 1 is a functional block diagram of the spectrum analyzer system. The Intermediate Frequency (IF) Input Signal Conditioner mixes, filters, and samples the analog input at the desired sampling frequency. The Input and Timing module collects sufficient data for a spectrum in one of the input buffers, selects the points for the first stages of the column transforms, and sends them to the Window module. The Input and Timing module also generates the synchronization signals for the pipeline control. The Window module multiplies the data by the coefficients of a window in order to suppress the sidelobes of the frequency response. The spectrum of the windowed data is computed by the matrix FFT using two separate FFT modules, one for the rows and one for the columns.

The 128-point column FFTs are 16-bit, fixed-point, radix-2 FFTs with two stages on a board. Each FFT stage performs a radix-2 FFT butterfly operation and then implements the delay for the next stage. The outputs from the column FFTs go into the matrix transpose memory, which selects the points for the row transforms. Each point is multiplied by the appropriate complex rotation factor called a "twiddle" factor prior to entering the second FFT module. The row FFTs are radix-4, floating-point FFTs with one stage per board. Both FFT modules are fully pipelined [1]. The matrix FFT was chosen because it allows small numbers of data points and twiddle factors to be resident on the FFT boards at the expense of performing the matrix transposes and twiddle factor multiplication. In order to compute the transform of a  $2N$ -long real sequence using an  $N$ -point complex FFT, the even points are loaded into the real components and the odd points into the imaginary components of an  $N$ -long complex sequence. The

transforms of the even and odd points are then formed from the symmetric and antisymmetric parts of the unscrambled transform, respectively, before the last radix-2 butterfly [2]. This final operation is called the real-adjust.

The FFT output data then go to the Power Accumulator which calculates the power of the complex spectrum samples and accumulates them on a frequency-by-frequency basis. There are actually two independent accumulators so that two separate data streams can be accumulated. The power spectra are then normalized using a predetermined baseline to remove frequency-dependent system gain. The normalized spectra then go through a matched filter designed to detect narrow-band sources as they are swept by a moving antenna during the SETI sky survey. The parameters of the filter can be adjusted based on noise statistics computed from the data spectra. The spectral processing for the graphics displays is performed by the Output Processor which can select a band of frequency points or pick peaks and averages of groups of contiguous frequency points.

The spectrum analyzer system is controlled by the microprocessor computer system whose primary function is data processing and graphics handling. The system controller and postprocessor is a 68020-based Masscomp 5600 computer system with high-resolution graphics and sufficient storage and processing power to control and configure the spectrum analyzer and to perform additional data processing on the output spectra. The computer system will generate graphics displays, archive data, provide antenna monitoring and control, and perform special-purpose processing. Examples of special-purpose processing include signal identification, signal tracking, and removal of undesired interfering signals.

The system controller also performs system tests and diagnostics. It commands the Executive board to generate test synchronization signals and transmit and receive test vectors through the stimulus and response buffers. The stimulus buffer contains 16 megabytes of read/write Masscomp-mapped memory—enough to hold  $2^{21}$  complex floating-point samples. Test input vectors generated by the microprocessor are written to the stimulus buffer at microprocessor speed and are then transferred to the Input and Timing module at the hardware processing speed. The Executive board can cycle through the input data to generate periodic stimuli. Output vectors return to the microprocessor via the response buffer, a ring buffer in which the last memory location is followed by the first one to form a continuous ring. Data streams into the response-buffer memory until a programmed trigger condition is met. After a trigger event, an additional specified number of data points are written, and then the data are available to be read. After reading the data, the user resets a bit to arm the memory and the cycle repeats. To check the sys-

tem performance, output vectors can be compared to expected response vectors. Since any module can be bypassed as the controller chooses, each module can be tested in isolation or with any combination of other modules. The data buses in the system are configured so that any board can be supplied with its full data-width stimulus and have the full width response captured in the response buffer. The widest data path consists of two complex numbers and two serial control lines. Complex numbers are pairs of floating-point numbers (R,I), each 32 bits wide; thus the widest data path is 130 bits wide. The floating-point representation is the IEEE 32-bit floating-point standard. The serial control lines carry the synchronization pulses.

### III. Synchronization and Configuration Control

The operational flexibility of the spectrum analyzer arises from two sources. The first source is the way the data flow is controlled using synchronization and framing signals; the second is the way parameters can be set to control the functions of the hardware modules. The spectrum synchronization signals, explained below, include the start-of-spectrum indicator, valid spectrum indicators, the *A/B* indicator, and accumulation frame triggers. The synchronization signals are generated by the Input and Timing module and sent down the processing pipe on two serial lines. Each module delays the synchronization signals by an amount equal to its pipe delay before sending the signals to the next module in the pipe.

The processing of a spectrum begins with data from one of the three 4-megabyte RAM buffers on the Input and Timing module. The current buffer is read out by the delay commutator which outputs pairs of complex data samples which are to be combined in the first butterfly stage of the transform. The start-of-spectrum indicator occurs every  $N$  complex samples, or every  $N/2$  complex pairs. Because of the periodicity of the delay commutator, the beginning of a new spectrum can occur only every  $N$  sample times. Thus if the sampling rate is less than the 40-MHz processing rate and the next buffer is not full when the commutator is reset to start a new spectrum,  $N$  sample times must be skipped before the buffer can begin to be processed. The first two time lines in Fig. 2 show the write and read times for a 64-MHz real sampling rate. Since the sampling rate is four-fifths the processing rate, after the start-up delay every fifth spectrum does not contain valid data, and a valid spectrum indicator does not appear in the third time line. The processing pipe cannot be stopped to wait for the buffer to fill; the modules thus continue cycling through their processing, but the absence of the valid spectrum indicator prevents the data from being added into the accumulations or entered into the spectrum filter computations.

Since two input channels can be in the pipe, there are two separate accumulator memories. The *A/B* indicator identifies which accumulation each valid spectrum belongs to. The lengths of the accumulations are controlled by the accumulation frame trigger which causes the current spectrum to be added into the appropriate accumulation; the next valid spectrum corresponding to that accumulation will then be deposited, not added, to the buffer. The triggers for the *A* and *B* accumulations are controlled independently. The bottom time line in Fig. 2 shows the *A* frame trigger for accumulations of three spectra when all the spectra are type *A*. The main use for the two separate channels is for Dicke switching. In Dicke switching the antenna switches from looking at a source for  $K_1$  spectra to looking at the background noise for  $K_2$  spectra. The spectrum analyzer controls the switch, discarding a specified number of spectra to allow synchronization with a pulsing signal or settling time for the Dicke switch. For other applications, such as dual-polarization processing, the two accumulations allow the analyzer to process the inputs from two IF stages with maximum bandwidths of 20 MHz. For dual-channel input the analyzer would alternate the spectra; thus the even-numbered spectra would belong to channel *A* and would be separate from the odd spectra which belonged to channel *B*.

In order to generate the correct synchronization signals, the Input and Timing module has a group of control bytes which specify the number of spectra in the *A* and *B* accumulations, the number of invalid spectra between switching accumulations, and the base-two logarithm of the spectrum length. There are also bits for a user-supplied sync code, for controlling the clock, and for forcing initialization of the analyzer. The other modules also contain one or more control bytes to pass module parameters and control bits such as bypass and read switches. Some of the controls allow multiple copies of one board design to be used for similar operations at different stages in the pipe. For example, there are two copies of the multiplier board, one to perform the window multiply on the input data and one to do the twiddle-factor multiply between the row and column transforms. The window is real, while the twiddle multiply is complex; thus one bit is used to select whether the board performs a real or a complex multiplication. For complex multiplications, another bit selects whether or not the result should be divided by two. To load the coefficient data onto the board, the operational/multibus switch is set to multibus mode, and the data words are transferred on the multibus.

The fixed-point FFTs are computed using four copies of one board design, and the floating-point FFTs use seven copies of a floating-point design. Each FFT board needs only one control byte to control which stage or stages of the FFT it computes. The fixed-point FFT boards contain two stages of a

radix-2 FFT, and the control byte contains the stage number,  $k$ , of the second stage, and two bypass bits, one for the first stage and one for the second. If the first stage is not bypassed, its stage number is  $k - 1$ . Since the floating-point boards contain only one stage of a radix-4 FFT, the control byte contains only one bypass bit and the stage number. Another function of the control bytes is to pass parameters unique to a board which program the operation of the board. The frequency-domain boards have such parameters, and they are described in Section VI.

All the boards have bypass switches which allow the input data to be passed directly to the output data lines. The only board whose output is of a different form from its input is the floating-point converter which converts 16-bit fixed-point numbers to 32-bit floating-point numbers. It therefore has two different bypass modes, one which passes 32-bit data to the output, and one which maps the 16-bit input data to the first 16 bits of the 32-bit output lines. The bypass switches are all set independently and allow great flexibility in the use of the analyzer. In test mode they allow every module to be tested independently or in combination with other modules in the pipe. In operational mode they allow the spectrum analyzer to be customized to fit the user's application. For example, FFTs shorter than the maximum length are implemented by skipping FFT stages using the bypass switches; the window stage can be skipped, and other processing stages can be skipped as desired.

The control bytes are in the memory-address space of the workstation; thus, when a value is assigned to a control-byte location, a multibus transfer to the analyzer hardware is initiated. This means that the system configuration can be altered in the length of time it takes to complete the multibus accesses. Since the data computed with the new configuration propagate down the pipe, in general there will be a delay before the output contains valid spectra.

#### IV. The Two-Million-Point DFT

The core of the spectrum analyzer is the two-million-point Matrix DFT which transforms the time-domain input signal into the frequency domain. A real window can be applied to the DFT input to shape the frequency response. The length of the transform,  $N$ , can be any power of two from  $2^{17} = 131,072$  to  $2^{21} = 2,097,152$ .

The windowed DFT of the input signal is:

$$Y_i(k) = \frac{1}{N} \sum_{n=0}^{N-1} x_i(n)w(n) \exp\left(-j2\pi \frac{nk}{N}\right) \quad (1)$$

$$0 \leq k \leq N - 1$$

where  $\{x_i(n)\}$  is the input data and  $\{w(n)\}$  are the window coefficients. The subscript indexes successive spectra or, equivalently, segments of  $N$  samples. Thus  $x_i(n)$  refers to the  $(iN + n)$ th time sample.

The spectrum computation is a Matrix FFT in which an  $N = LM$ -point DFT is computed using  $L$ - and  $M$ -point FFTs and a complex multiply stage. The Matrix algorithm is derived in detail in [1]. The central idea is that the  $N$ -long input sequence is decomposed into an  $L$  by  $M$  matrix such that:

$$y_i(l, m) = y_i(ML + m) \quad (2)$$

$$0 \leq l \leq L - 1, \quad 0 \leq m \leq M - 1$$

where  $y_i(n) = x_i(n)w(n)$  for the  $i$ th spectrum. The output transform sequence is recomposed by the mapping:

$$Y_i(s, r) = Y_i(Lr + s) \quad (3)$$

$$0 \leq s \leq L - 1, \quad 0 \leq r \leq M - 1$$

Rewriting Eq. (1) in terms of the matrix indices yields:

$$Y_i(s, r) = \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} y_i(l, m) \exp\left(-j2\pi \frac{(ML + m)(Lr + s)}{ML}\right)$$

$$= \sum_{m=0}^{M-1} \exp\left(-j2\pi \frac{mr}{M}\right) \exp\left(-j2\pi \frac{ms}{ML}\right)$$

$$\times \sum_{l=0}^{L-1} y_i(l, m) \exp\left(-j2\pi \frac{ls}{L}\right) \quad (4)$$

The rightmost summation is a DFT of length  $L$  on the  $m$ th column. The result will be called  $q_i(s, m)$ . The summation over  $m$  is an  $M$ -point DFT on the  $s$ th row of the matrix  $\exp[(-j2\pi)(ms/LM)] q_i(s, m)$ . The complex phasors  $\exp[(-j2\pi)(ms/LM)]$  are called the twiddle factors.

The advantage of the Matrix FFT method over a straight Decimation-in-Frequency (DIF) or Decimation-in-Time (DIT) FFT computation [1] is that by decomposing the  $N$ -point transform into short  $M$ - and  $L$ -point transforms, the long delay memories and twiddle-factor tables are not resident on the FFT boards but are concentrated into one large  $N$ -point matrix transpose buffer and one large twiddle-factor table.

In the JPL spectrum analyzer the column transforms are computed via radix-2, DIF, fixed-point FFTs. In a radix-2

algorithm the FFT is computed in  $K = \log_2 L$  stages. Each stage consists of a "butterfly" operation on  $L/2$  pairs of complex points. The butterfly operation

$$z_k(l) = \frac{z_{k-1}(l) + z_{k-1}(l + 2^{K-k})}{2} \quad 0 \leq l \leq L - 1$$

$$z_k(l + 2^{K-k}) = \frac{\{z_{k-1}(l) - z_{k-1}(l + 2^{K-k})\} W_L^{l2^{k-1}}}{2} \quad 1 \leq k \leq K \quad (5)$$

takes the sum and difference of the complex pairs and then multiplies the difference by the appropriate twiddle factor. Dividing the results by two ensures that the maximum magnitude of the numbers does not increase by a factor of two at every stage. At the  $k$ th stage,  $1 \leq k \leq K$ , the pairs for each butterfly operation are the results from the previous stage taken  $2^{K-k}$  apart,  $z_{k-1}(l)$  and  $z_{k-1}(l + 2^{K-k})$ , and the input to the first stage is the windowed data from the  $m$ th column,  $z_0(l) = x_i(ML + m)w(ML + m)$ . The output from the last stage is  $q_i(U(l)M + m) = z_K(l)$ , where  $U(l)$  is the bit reversal of  $l$ .

The output from the fixed-point column transforms enters the matrix-transpose memory which unscrambles the columns and selects the data quadruples for the row FFTs. Before the row transforms, the points are multiplied by the appropriate twiddle factor and converted from 16-bit, two's-complement, fixed point to IEEE 32-bit floating point. The row transforms are radix-4, floating-point FFTs. In the radix-4 algorithm the FFT is computed in  $K = \log_4 M$  stages in which each stage implements a 4-point DFT followed by twiddle-factor multiplications:

$$\begin{bmatrix} u_k(m) \\ u_k(m + 4^{K-k}) \\ u_k(m + 2 \times 4^{K-k}) \\ u_k(m + 3 \times 4^{K-k}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} z_{k-1}(m) \\ z_{k-1}(m + 4^{K-k}) \\ z_{k-1}(m + 2 \times 4^{K-k}) \\ z_{k-1}(m + 3 \times 4^{K-k}) \end{bmatrix}$$

$$z_k(m) = u_k(m)$$

$$z_k(m + p \times 4^{K-k}) = u_k(m + p \times 4^{K-k}) W_M^{mp4^{k-1}} \quad 1 \leq p \leq 3 \quad (6)$$

The input to the first stage consists of the twiddled points from the  $s$ th row,  $z_0(m) = \exp [(-j2\pi)(ms/LM)] q_i(s, m)$ . The output from the last stage is  $Y_i(U_4(m)L + s) = z_K(m)$ , where  $U_4(m)$  is the 4-based digit reversal of  $m$ .

Only two unique board designs are needed for the FFT computations: one for the fixed-point, radix-2 transforms, and one for the floating-point, radix-4 transforms. Although stage  $k + 1$  in a radix- $n$  FFT requires only  $1/n$ th the delay and coefficient memory of stage  $k$ , it is more economical to design, test, and maintain the boards if they are all exact replicas with the maximum memory requirements. Figure 3 shows a block diagram of a fixed-point FFT stage. On each board there are two FFT stages, each with an Arithmetic Unit (AU) to perform the complex butterfly, a PROM coefficient memory for the twiddle factors, and an SRAM delay memory with a cross-switch. In addition, on each board there is a synchronization generator unit, a clock module, a local bus module, and a thermal protection module.

The complex addition of Eq. (5) is done as a 17-bit add with "OR rounding" carried out when the bit shift is done for the division by two. The complex subtraction uses 16-bit subtracters which require the input data to have a magnitude less than  $1/2$ . To avoid overflow conditions in the subtracters, the Input buffer is wired to ensure that the 8-bit data has a magnitude less than  $1/2$ , and the window multiplication and FFT butterflies are designed not to increase the maximum magnitude. Since the window coefficients have a maximum magnitude of one, the real window multiplication will have a resultant less than  $1/2$ , and the divide-by-two in the FFT stages guarantees that the results maintain a maximum magnitude of  $1/2$ . The twiddle-factor complex multiply is performed using Multiplier-Accumulators (MACs), dividing by two in the process. The twiddle factors are stored in PROM memory as  $\cos(\theta) - j \sin(\theta)$ . There is a separate set of 128 sine/cosine pairs for each possible stage number; the appropriate set is read sequentially and repetitively. Although each butterfly stage increases the maximum signal level by at most 1 bit, with 8-bit input samples and seven stages, the 16-bit arithmetic has sufficient dynamic range [4].

The output from the AU enters the cross-switch which changes the order of the data, choosing the correct data pairs for the next FFT stage. The input pairs to the cross-switch at the  $k$ th stage are  $2^{K-k}$  apart; the pairs' outputs to the next stage are  $2^{K-k-1}$  apart. This operation is carried out using a two-input, two-output multiplexer and a read-modify-write memory which outputs a location and then stores new data in the same location. The memories act as  $2^{K-k-1}$  delay memories, while the multiplexer either passes data straight through or switches the data paths. The purpose of the sync code generator unit is to delay the sync code lines to match

the data delay appropriate for the stage number. The delay is calculated via a look-up table using the stage number as the address.

Each floating-point board contains one stage of a radix-4 DIF FFT. The AUs have eleven 50-nsec IEEE 32-bit floating-point adders and six 50-nsec IEEE 32-bit floating-point multipliers to perform the radix-4 butterflies. The 32-kilobyte twiddle-factor memory is made up of  $8k \times 8$  PROMs, and the cross-switch uses  $8k \times 8$  static RAMs. The output from the floating-point FFTs is digit reversed, and the final spectrum is formed in the Real-Adjust stage.

## V. The Real-Adjust Stage

A 40-MHz IF bandwidth can be sampled using a complex demodulator sampling at 40 million complex samples per second, or using a real demodulator sampling at 80 million real samples per second. The  $N$ -point DFT of the complex sampled signal would have  $N$  independent complex points spaced  $40 \text{ MHz}/N$  apart. For equivalent resolution, a  $2N$ -point transform would have to be computed on the real sampled signal. Since the spectrum of the real signal is conjugate symmetric with  $Y(2N - n) = Y^*(n)$ ,  $0 \leq n \leq N$ , only the first  $N + 1$  points actually have to be computed. The first  $N + 1$  points of the  $2N$ -point transform of the real signal  $y(m)$ ,  $0 \leq m \leq 2N - 1$ , can be computed using an  $N$ -point complex DFT by the following algorithm:

$$z(n) = y(2n) + jy(2n + 1), \quad 0 \leq n \leq N - 1 \quad (7)$$

Let Eq. (7) be the input to an  $N$ -point transform. Then the output  $Z(k)$ ,  $0 \leq k \leq N - 1$ , is the sum of the transforms of the real sequence  $y(2n)$  and the imaginary sequence  $y(2n + 1)$ ,  $0 \leq n \leq N - 1$ . Since a real sequence has a conjugate symmetric spectrum and an imaginary sequence has a conjugate antisymmetric spectrum, the transforms of the even and odd points can be recovered from  $Z(k)$  [2]:

$$\begin{aligned} Y_E(k) &= \frac{1}{2} \{Z(k) + Z^*(N - k)\} \\ Y_O(k) &= \frac{-j}{2} \{Z(k) - Z^*(N - k)\}, \quad 0 \leq k \leq N - 1 \end{aligned} \quad (8)$$

where  $Z(N) = Z(0)$ . From the transforms of the even and odd points, the full transform is:

$$Y(k) = \{Y_E(k) + Y_O(k)W_{2N}^k\}$$

$$Y(N-k) = \{Y_E^*(k) - Y_O^*(k)W_{2N}^{*k}\}, \quad 0 \leq k \leq N \quad (9)$$

Note that this full real adjust would generate  $N+1$  complex numbers from the  $N$  complex inputs, yet the pipe can only accommodate  $N$  complex numbers. To avoid this problem  $Y(N)$  is not computed, since  $Y(N)$  would not have been computed if complex demodulation had been used. In Eqs. (8) and (9), the point  $Y(0)$  is computed only from  $Z(0)$  and  $Y(N/2)$  only from  $Z(N/2)$ . These two points are therefore handled together as the first data pair with the operation:

$$\begin{aligned} Y(0) &= \text{Re}\{Z(0)\} + \text{Im}\{Z(0)\} \\ Y(N/2) &= Z^*(N/2) \end{aligned} \quad (10)$$

## VI. Frequency-Domain Processing Modules

The real-time, frequency-domain processing modules automate the detection of signals in the spectra and reduce the volume of output data. These modules include the Power Accumulator, the Baseline module, the SETI Filter, and the Output Processor. Since the spectrum analyzer under construction is a prototype design for a segment of the analyzer to be used in the sky survey portion of NASA's SETI program, these modules were designed to meet the requirements for detecting narrowband signals as an antenna scans the sky. Essentially the goal of the survey is to search the entire celestial sphere in the microwave region from 1 to 10 GHz for signals as narrow as a few hertz. The entire survey is expected to take about five years to complete and will use the radio telescopes of the DSN and other large antennas. Since the survey involves an ambitious amount of data and processing, the spectrum analyzer is designed to perform the signal detection steps which must be applied to every spectral point in the pipe. These steps include accumulating the power at each frequency, removing frequency-dependent system gain, and using a matched filter to detect stationary celestial sources scanned by the antenna beam. Since the points which pass the matched-filter threshold test contain potentially interesting signals, they are sent to the workstation for postprocessing.

The first frequency-processing module is the Power Accumulator which computes the power of each of the complex points output by the FFT and accumulates the power values in one of the channel-accumulation memories. The length of the accumulation,  $L$ , can be from 1 to  $2^{24}$  spectra. The Baseline module then removes the frequency-dependent system

gain from the accumulated spectrum by multiplying each channel by a precomputed inverse baseline value. The baseline,  $B(n)$ , is computed by the workstation using spectra from an input noise source. The baseline can be updated while the antenna is being repositioned between scans. The accumulated, baselined output,  $S_k(n)$ , is:

$$S_k(n) = \frac{1}{B(n)} \sum_{m=L(k-1)}^{Lk-1} |Y_m(n)|^2, \quad k \geq 0 \quad (11)$$

The baselined spectra enter the SETI Filter module which is designed to detect narrowband signals and report the frequency, bandwidth, and accumulated power of these signals to the workstation. The SETI Filter module contains a five-coefficient Finite Impulse Response (FIR) filter which takes a running average of each channel across successive accumulated spectra. This spectrum filter can be a matched filter for any desired single-frequency time signature of duration less than or equal to five accumulation intervals. The filter output

$$F_k(n) = \sum_{m=0}^4 \alpha_{k,m}(n) S_{k-m}(n) \quad (12)$$

where the filter weights  $\alpha_{k,m}(n)$ ,  $0 \leq m \leq 4$  are equal to the desired time signature. The filtered data then goes to the Threshold board which taps the data and locates regions with power above a computed frequency-dependent threshold. The location, the number of frequency points in the region, and the total power in the region are computed and stored in an event memory which can be accessed by the workstation. The Threshold board contains a bit mask which can be used to force frequency ranges known to be contaminated with RFI to be ignored.

In order to allow for a varying noise temperature across the 40-MHz signal bandwidth, the spectrum is divided into a maximum of 128 subspectra, each with its own threshold and filter weights which depend on an estimate of the noise temperature in the subspectrum. This estimate is proportional to the power in the  $N$ th dimmest spectral bin, where  $N$  is a control parameter. The filter weights are equal to their initial static values, or the initial values divided by the estimated noise temperature. The threshold is equal either to an input constant  $\theta$ , or to  $\theta$  times the sum of the filter weights. The  $N$ th dimmest spectral bin for each subspectrum is determined with 24-bit accuracy by using a coarse-resolution histogram of the twelve most significant bits, determining the bin where the  $N$ th dimmest power lies, fixing these twelve most significant bits, and then doing a histogram of the next twelve significant bits.

The purpose of the matched filter is to integrate the power received at each frequency as the antenna beam sweeps across a source in order to use all the information from the source. Figure 4 depicts a one-dimensional slice of the circularly symmetric expected response at a single frequency. The combination of accumulating and then filtering results in a weighting function which is a step-like approximation to the bell-shaped response. Combining five accumulations in this fashion as the antenna moves one Half-Power Beamwidth (HPBW) results in an SNR loss of 0.25 to 0.3 dB [5]. Figure 5 depicts an oval antenna scan pattern with circles showing half-power beam areas. The filtering implemented by the hardware integrates only along the direction of the antenna scan. To approach uniform spatial sensitivity, the information from adjacent scans should be combined as well. Adjacent scans are expected to be separated by one HPBW, as shown in Fig. 5; thus, sources positioned at the edge between two scans would register in both scans, 3 dB lower than if the source were centered in one of the scans.

In order to combine the data from adjacent scans optimally, all the spectral data generated between successive passes past a position would have to be stored; for a reasonable antenna scan pattern, this results in an unreasonable amount of data storage. By sending only the spectral points which pass a threshold, the amount of data stored can be reduced to a fraction of the total with very little effect on the probability of missing a signal. For example, setting the threshold to retain only 1 percent of the data reduces the probability of missing a narrowband signal by only one or two percent, but dramatically reduces the maximum required storage. The combination of data from scan to scan is then done by software in the workstation.

In addition to the matched-filter output, the workstation receives data from the Output Processor which controls the amount and form of the spectrum analyzer output to be used for spectrum displays. The Output Processor has the ability to store both the peaks and averages of contiguous groups of spectral bins called "superbins" in a set of Multibus-mapped

buffers. The number of spectral bins per superbin is a programmable power of 2, and the first superbin may start at an arbitrary programmable location. The output of this data reduction process goes into two double buffers, one for the peaks and one for the averages of each superbin. The buffers can each contain a maximum of 1024 IEEE 32-bit floating-point numbers. If the user-supplied parameters result in more than 1024 superbins, the extra bins are not calculated.

## VII. Conclusion

The wideband spectrum analyzer being developed is a powerful tool for real-time frequency-domain analysis of widebandwidth input signals. By using the signal processing boards to average and threshold the spectra, the uninteresting data can be reduced and signals of potential interest, such as continuous-wave or narrowband signals, can be retained. The system design principles were developed from experience with other large real-time systems, including a 20-MHz-bandwidth, 65,536-point spectrum analyzer completed in 1979 [6] and a real-time signal processing system for synthetic aperture radar containing four 20-MHz-wide, 16,384-point FFTs for performing convolutions. One of the design goals was to make the hardware as flexible as possible within the constraints of the pipeline architecture while keeping the size down to one rack of equipment. The sampling rate, the size of the transform, and the time-domain window are the only parameters needed to control the bandwidth and resolution of the output spectra. Most of the programmable parameters control the real-time signal processing boards where flexibility is needed in order to allow specification of a wide range of signal detection algorithms. The 40-MHz,  $2^{21}$ -channel analyzer is a stand-alone system which is the prototype for an analyzer with a bandwidth of 300 MHz. Some of the expected applications for both the 40-MHz and 300-MHz designs, in addition to the SETI sky survey, include radio astronomy spectral analysis and continuum observations; wideband, high-resolution frequency analysis of receiver systems in the DSN; detection of radio-frequency interference; and spacecraft telemetry acquisition and analysis.



## References

- [1] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., pp. 371–377 and 602–613, 1975.
- [2] J. W. Cooley, P. A. W. Lewis, and P. D. Welch, “The Fast Fourier Transform Algorithm: Programming Consideration in the Calculation of Sine, Cosine, and Laplace Transforms,” *J. Sound Vib.*, vol. 12, pp. 315–337, July 1970.
- [3] E. T. Olsen, A. Lokshin, and S. Gulkis, “An Analysis of the Elements of an All Sky Survey,” presented at IAU Symposium No. 112, Boston, Massachusetts, June 1984.
- [4] E. H. Satorius, M. J. Grimm, G. A. Zimmerman, and H. C. Wilck, “Dynamic Range Constraints for Large Size, Hybrid Fixed/Floating Point FFTs,” presented at the 20th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, California, November 10–12, 1986.
- [5] J. Solomon, M. P. Quirk, E. T. Olsen, and W. Lawton, “A Signal Detection Strategy for the SETI All Sky Survey,” *TDA Progress Report 42-83*, vol. July–September 1985, Jet Propulsion Laboratory, Pasadena, California, pp. 191–208, November 15, 1985.
- [6] R. F. Emerson, “Biplex Pipelined FFT,” *DSN Progress Report 42-34*, vol. May–June 1976, Jet Propulsion Laboratory, Pasadena, California, pp. 54–59, August 15, 1976.
- [7] G. A. Morris and H. C. Wilck, “JPL  $2^{20}$  Channel 300 MHz Bandwidth Digital Spectrum Analyzer,” presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing, Tulsa, Oklahoma, 1978.

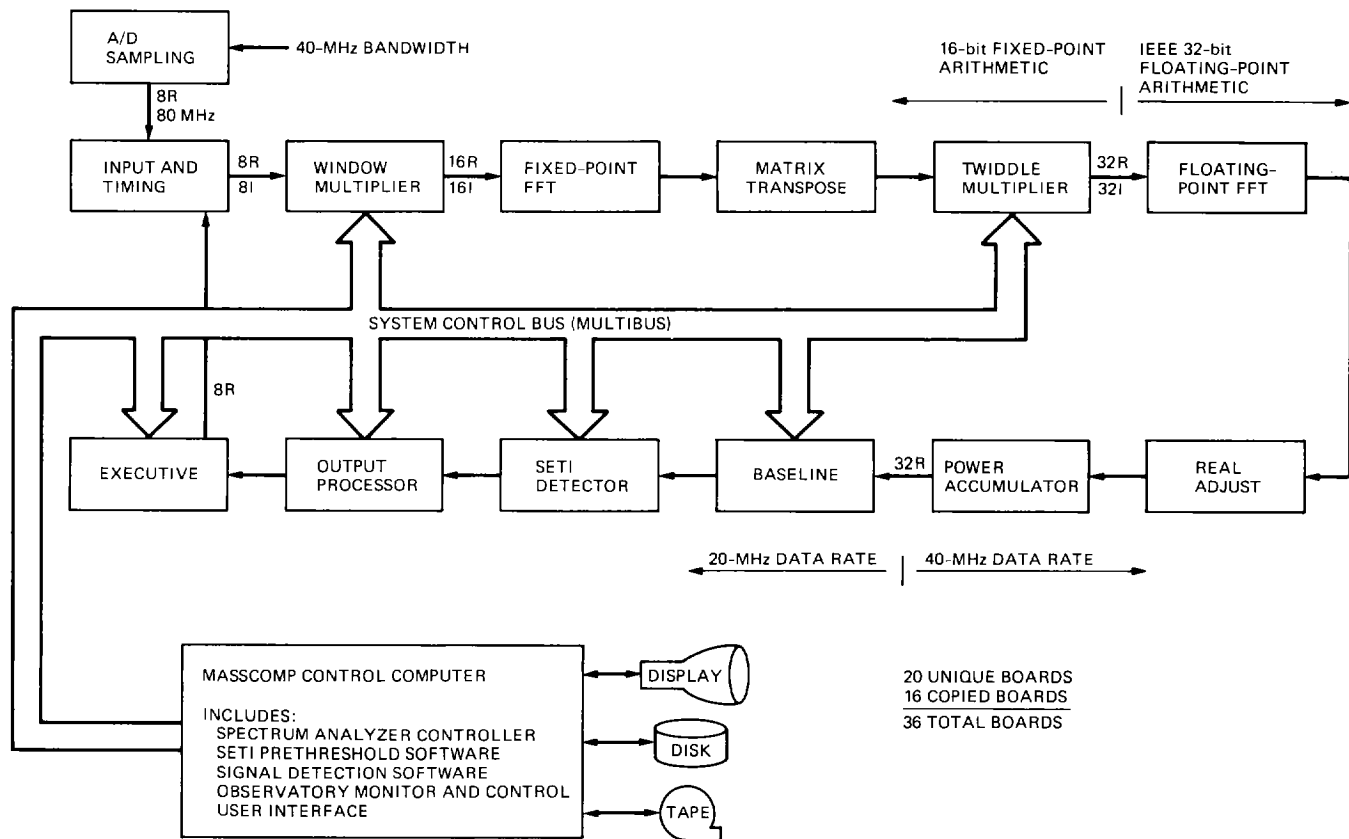


Fig. 1. The DSN-SETI spectrum analyzer system

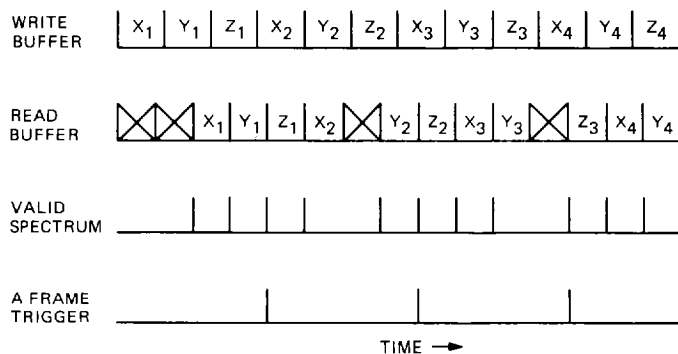


Fig. 2. Input buffer write and read cycles, valid-spectrum indicator, and accumulation-frame trigger for a 32-MHz sampling rate with three accumulations per frame

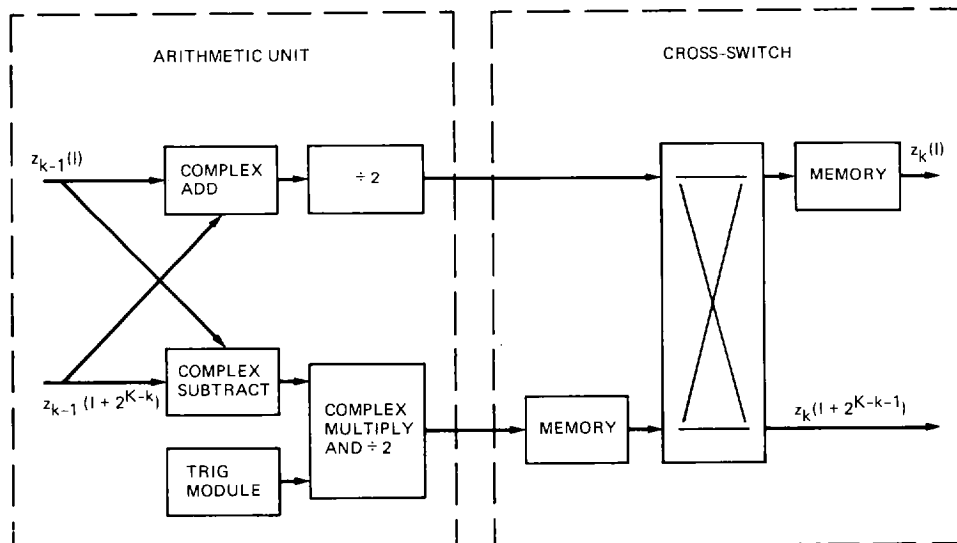


Fig. 3. Block diagram for a radix-2 fixed-point FFT stage

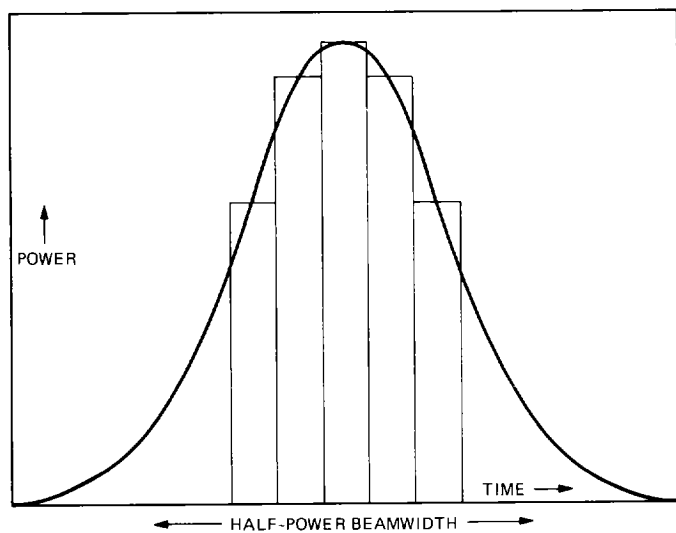


Fig. 4. Response to a point source transiting the antenna beam. Matched-filter values result in a staircase approximation to the response.

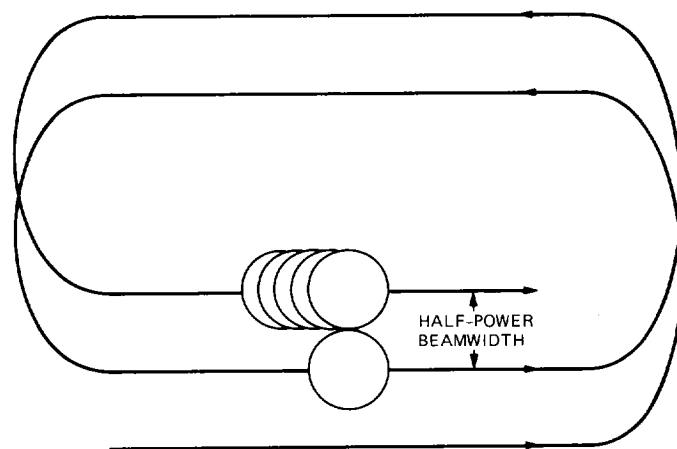


Fig. 5. Oval antenna scan pattern. Circles represent half-power beam areas.